Executive Summary

Accelerating Security Applications with Intel® Multi-core Processors

Key findings

- Software developers may achieve significant performance improvements enabled by Intel® multi-core processors without extensive program modifications.
- Intel has demonstrated a 6x performance increase with minor modifications to a singe-threaded security application for use on four processor cores.
- Through "pipelining" and "flow-pinning," a near 2x improvement in L2 cache efficiency was achieved.

Summary

The transition to multi-core processors offers flexibility for development and optimization of higher performance applications. Multi-core architectures can significantly improve program flow so that cache memory associated with each execution core is used more effectively. With multiple caches, optimizing data locality is possible, driving higher cache-hit rates and improved overall application performance.

Intel demonstrated over 6x performance improvement of an intrusion detection and prevention security application running on four processor cores (dual-core, dual-processor) – without threading the application. Such supra-linear performance improvements offer new opportunities for developers to inexpensively improve their applications while also achieving faster time-to-market.

6X Performance Gain With Pipelining and Flowpinning

The performance of most processors is highly dependent upon the availability of data and instructions to the execution unit. Optimizing cache efficiency, measured as cache hit-rate, lowers effective memory latency and improves performance.

The cache hit-rate often correlates to the application program's locality of reference, or the degree to which a program's memory accesses are limited to a relatively small number of addresses. A program that accesses a large amount of data from scattered addresses is less likely to use cache memory efficiently. Distributing TCP reassembly among several execution cores, it is possible to improve the locality of reference in a packet processing application by "pinning" individual TCP flows to a particular execution core, known as flow-pinning. Functional pipelining is another technique that sub-divides application software into multiple sequential stages and assigns these stages to dedicated, more static selection of execution units.

Pipelining allows parts of applications to run more efficiently on separate cores, while flow-pinning locks each TCP flow to a particular core. These techniques enable higher reference of locality and better cache efficiency, permitting TCP flows to process much faster with a large number of connections.

Pipelining and flow-pinning are applicable to many packet processing applications found in networking and communications, offering significant benefits to developers and their customers.

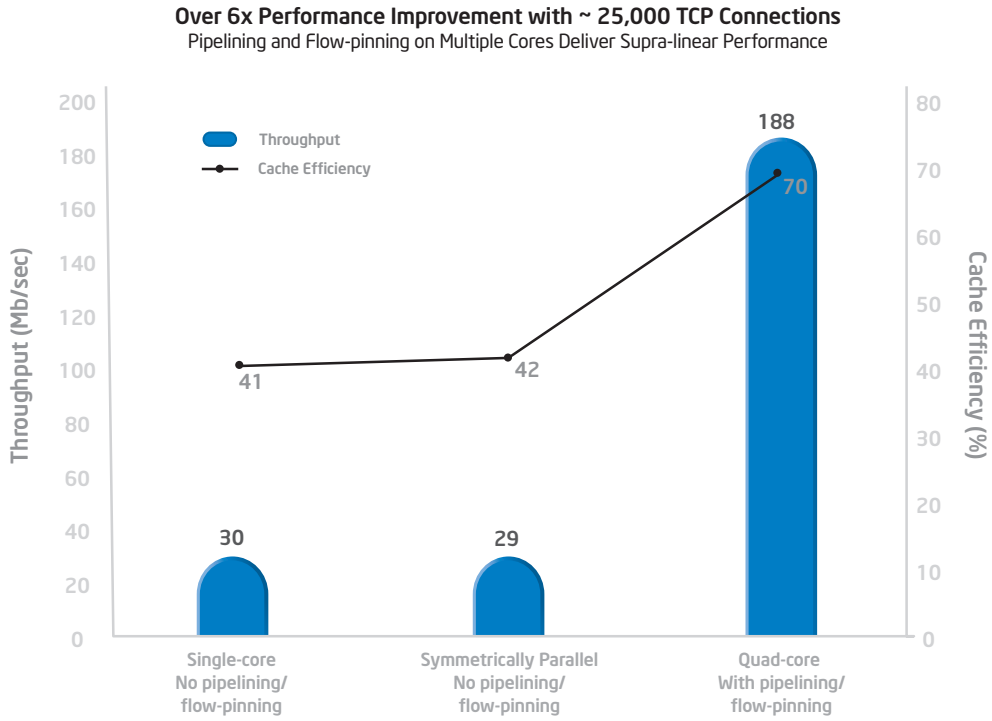

Figure 1. Performance comparisons

The data file used to evaluate throughput performance represents network traffic. Actual results may vary. Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations www.intel.com/performance/resources/limits.htm

Intel applied pipelining and flow-pinning techniques to Snort*, an open-source intrusion detection and protection security application, and tested it in three configurations on a dual-core dual-processor system using Dual-Core Intel® Xeon® processor LV 2.0 GHz to evaluate the effects of their modifications. The first unmodified configuration ran Snort and all TCP flows on just one core of the system, leaving the other three cores largely idle. The second configuration ran an instance of Snort and multiple TCP flows in parallel on each of the four cores. The last configuration used one core to assign, or pin, all packets of a TCP flow to a core, using multiple cores over many flows.

Using both pipelining and flow-pinning techniques in a configuration exercised with approximately 25,000 distinct packet flows, or connections, a 6.2x performance gain was achieved. (see Figure 1).

For more information

For more information about these techniques and the results of this testing, download the Intel® white paper, "Supra-linear Packet Processing Performance with Intel® Multi-core Processors," available at www.intel.com/technology/advanced_comm/311566.htm.

Two Dual-Core Intel® Xeon® processors LV 2.0 GHz were used in this project. Each processor incorporates two 2.0 GHz cores, with 2 MB of dynamically assigned L2 Cache, at a Thermal Design Point (TDP) of 31 Watts (a total of 62 W TDP for the two processors in this experiment). www.intel.com/design/intarch/dualcorexeon/overview.htm

Snort is an open-source system utilizing a rule-driven language that combines the benefits of signature, protocol and anomaly-based inspection methods. Please see www.snort.org for further details on this security application.

Information regarding third-party products is provided solely for educational purposes. Intel is not responsible for the performance or support of third-party products and does not make any representations or warranties whatsoever regarding quality, reliability, functionality, or compatibility of these devices or products.

*Other names and brands may be claimed as the property of others.

Printed in USA 0706/CAM/OCG/XX/PDF ♻ Please Recycle 314312-001US